

# IND320 CA1 Ivar Eftedal

## Use of AI

There was very little use of AI in this project. The only thing i needed help with the ai for was to make the days into the dateTime format so that i could select based on months. Other than that everything is done by reading documentaion and googling.

## Log of compulsory work

- The compulsory work gave me a nice base for creating an interactive dashboard to display some weather data. I learned some of the basics and fundamentals of streamlit. I was able to get most of the code working without too many issues. My biggest challenge was splitting the dates into months. I had trouble parsing the format into a date format. But other than that things went fairlt smooth.
- In order to learn what i needed to do, I spent most of my time reading the documentation of streamlit. I also spent some time reading the documentaion of pandas, as i do not have too much experience with this framework. For the more problematic challenges i faced, i went to stack overflow for inspiration on how to find solutions and examples of problems that were similar to mine.

## Experience with jupiter notebook and streamlit

- Working with streamlit was pretty nice. The documentation was decent, and easy to follow. I really liked the fact that they included examples in the documentation. This made it much easier to implement what i wanted. It was easy to get up and running, and I did not face any issues with streamlit.
- I did however not enjoy working with jupiter notebooks. I feel like it is clunky and unnecessary. I prefer working in normal files and just creating functions instead of code blocks. Since you can run cells out of order, this can lead to code working when it maybe shouldnt. When trying new stuff it might seem to work, but then when restarting it does not work anymore. This makes debugging and following proper structure challenging.
- I hope that jupiter notebooks will not play a big role in this course. I would rather prefer to document my code well in the normal files, as well as writing other important stuff in the README.md file or a text file or something else.
- With that beig said i think this first assignment was great, and gave me a solid intuition for quickly deploying a simple dashboard.

## Links:

- [GitHub](#)
- <https://github.com/ioeftedal/ind320ioeftedal>
- [Streamlit](#)
- <https://ind320ioeftedal.streamlit.app>

## Getting into the code:

Start by installing dependencies

Import the installed dependencies

```
import pandas as pd
import matplotlib.pyplot as plt
```

Read the provided csv file

```
df = pd.read_csv("../data/open-meteo-subset.csv")
```

Print all the data in the dataframe

```
print(df)
```

	time	temperature_2m (°C)	precipitation (mm)	\
0	2020-01-01T00:00	-2.2	0.1	
1	2020-01-01T01:00	-2.2	0.0	
2	2020-01-01T02:00	-2.3	0.0	
3	2020-01-01T03:00	-2.3	0.0	
4	2020-01-01T04:00	-2.7	0.0	
...	...	...	...	
8755	2020-12-30T19:00	-6.5	0.0	
8756	2020-12-30T20:00	-7.0	0.0	
8757	2020-12-30T21:00	-7.5	0.0	
8758	2020-12-30T22:00	-7.2	0.0	
8759	2020-12-30T23:00	-7.5	0.0	

	wind_speed_10m (m/s)	wind_gusts_10m (m/s)	wind_direction_10m (°)
0	9.6	21.3	
284			
1	10.6	23.0	
282			
2	11.0	23.5	
284			
3	10.6	23.3	
284			
4	10.6	22.8	
284			
...	...	...	...
..			
8755	1.6	4.0	
58			
8756	1.2	3.4	
50			
8757	1.1	2.4	
22			

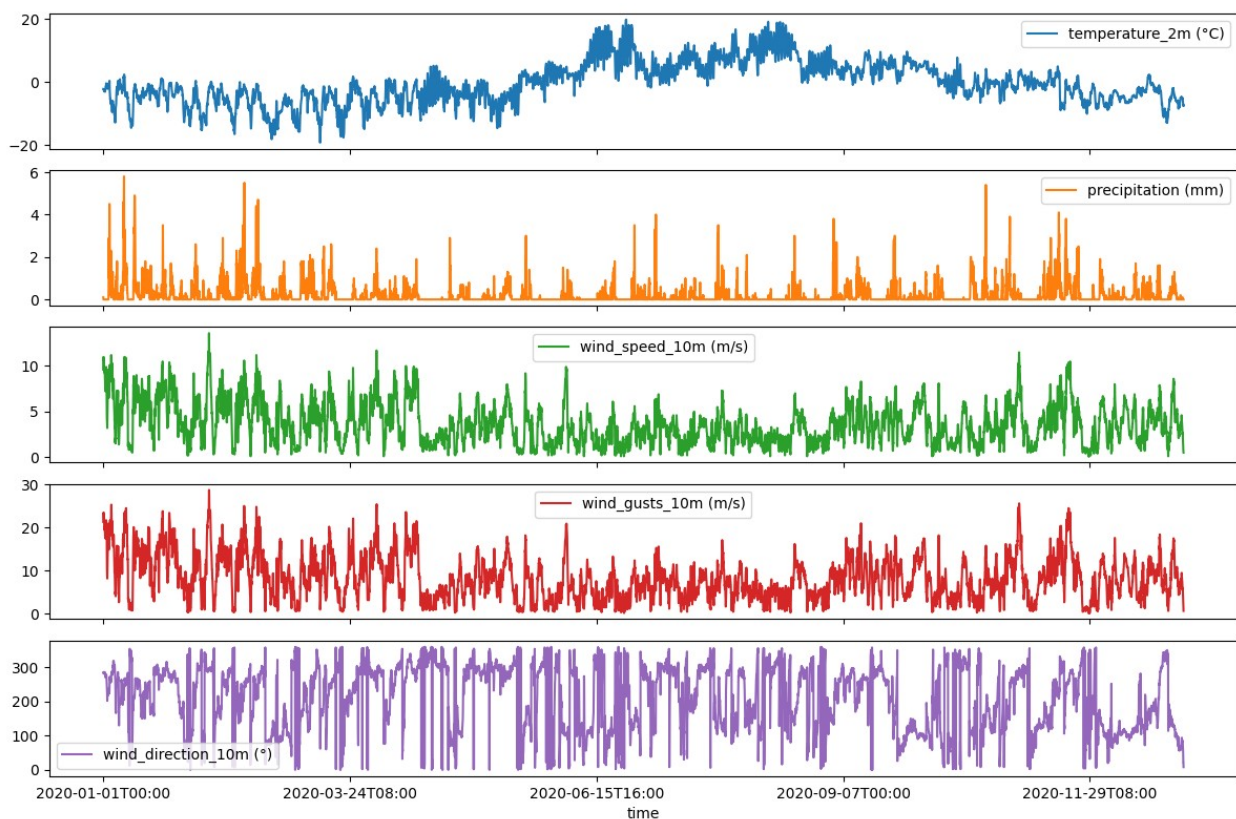
8758	0.6	1.6
22		
8759	0.5	0.7
8		

[8760 rows x 6 columns]

Print each column separately

```
df.plot(subplots=True, figsize=(12, 8))
```

```
plt.tight_layout()
plt.show()
```



Print all columns together

```
fig, ax1 = plt.subplots(figsize=(10, 6))
```

```
ax1.plot(df['time'], df['temperature_2m (°C)'], color = 'C0')
ax1.set_xlabel('Time')
ax1.set_ylabel('temperature_2m (°C)', color = 'C0')
ax1.tick_params(axis = 'y', labelcolor = 'C0')
```

```
ax2=ax1.twinx()
ax2.spines["right"].set_position(("axes", 1))
```

```

ax2.plot(df['time'], df['precipitation (mm)'], color = 'C1')
ax2.set_ylabel('precipitation (mm)', color = 'C1')
ax2.tick_params(axis = 'y', labelcolor = 'C1')

ax3=ax1.twinx()
ax3.spines["right"].set_position(("axes", 1.1))
ax3.plot(df['time'], df['wind_speed_10m (m/s)'], color = 'C2')
ax3.set_ylabel('wind_speed_10m (m/s)', color = 'C2')
ax3.tick_params(axis = 'y', labelcolor = 'C2')

ax4=ax1.twinx()
ax4.spines["right"].set_position(("axes", 1.2))
ax4.plot(df['time'], df['wind_gusts_10m (m/s)'], color = 'C3')
ax4.set_ylabel('wind_gusts_10m (m/s)', color = 'C3')
ax4.tick_params(axis = 'y', labelcolor = 'C3')

ax5=ax1.twinx()
ax5.spines["right"].set_position(("axes", 1.3))
ax5.plot(df['time'], df['wind_direction_10m (°)'], color = 'C4')
ax5.set_ylabel('wind_gusts_10m (m/s)', color = 'C4')
ax5.tick_params(axis = 'y', labelcolor = 'C4')

ax1.set_title('Weather Data In 2020')
fig.show()

/var/folders/sz/3yw64x157p3b5x22430zk7p000000gn/T/
ipykernel_57373/564956209.py:33: UserWarning: FigureCanvasAgg is non-
interactive, and thus cannot be shown
fig.show()

```

